

Amendment to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method of operating a multithreaded parallel processor comprising:

directing the processor having a plurality of microengines to swap a currently running context, corresponding to a first thread, in a specified microengine ~~out to memory~~ to let another context, corresponding to a different thread, execute in that microengine and cause a different context and associated program counter to be selected.

2. (Currently amended) The method of claim 1 wherein the directing the processor wakes up the swapped out context when a ~~specified~~ signal specified in a context-swap program instruction is activated.

3. (Previously presented) The method of claim 2 wherein the signal is specified as a parameter in the directing and specifies an occurrence of an event.

4. (Previously presented) The method of claim 3 wherein the parameter specifies "sram Swap", and the directing swaps out a current context and wakes it up when the thread's SRAM signal is received.

5. (Previously presented) The method of claim 3 wherein the parameter specifies "sdram Swap," the directing will swap out a current context and wakes it up when the thread's SDRAM signal is received.

6. (Previously presented) The method of claim 3 wherein the parameter specifies "FBI" will swap out a current context and wakes it up when the thread's FBI signal is received indicating that an FBI CSR, Scratchpad, TFIFO, or RFIFO operation has completed.

7. (Previously presented) The method of claim 3 wherein the parameter specifies "seq_num1_change/seq_num2_change", which swaps out a current context and wakes it up when a value of the sequence number changes.

8. (Previously presented) The method of claim 3 wherein the parameter specifies "inter_thread" which swaps out a current context and wakes it up when the thread's interthread signal is received.

9. (Previously presented) The method of claim 3 wherein the parameter specifies "voluntary" which will swap out a current context if another thread is ready to run, and if the thread is swapped, the swapped thread is automatically re-enabled to run at some subsequent context arbitration point.

10. (Previously presented) The method of claim 3 wherein the parameter specifies "auto_push" which swaps out a current context and wakes it up when SRAM transfer read register data has been automatically pushed by a FBus interface.

11. (Previously presented) The method of claim 3 wherein the parameter specifies "start_receive" that swaps out a current context and wakes it up when new data in a receive FIFO is available for this thread to process.

12. (Previously presented) The method of claim 3 wherein the parameter specifies "kill" which prevents a current context or thread from executing again until an appropriate enable bit for the thread is set in a CTX_ENABLES register.

13. (Previously presented) The method of claim 3 wherein the parameter specifies "pci" which swaps out a current context and wakes it up when a PCI unit signals that a DMA transfer has been completed.

14. (Previously presented) The method of claim 3 wherein directing further comprises:

an optional_token "defer one" which specifies that one instruction will be executed after this reference before the context is swapped.

15. (Currently amended) A method of operating a multithreaded parallel processor ~~comprises, the method comprising:~~

evaluating a specified parameter to determine a state of an executing context process corresponding to a first thread; and

performing a swapping operation to cause a different context and associated program counter, corresponding to a different thread, to be selected in accordance with the value of the evaluated specified parameter.

16. (Currently amended) The method of claim 15 wherein performing swaps a currently running context in a specified microengine ~~out to memory~~ to let another context execute in that microengine.

17. (Original) The method of claim 15 wherein the parameter specifies an occurrence of an event.

18. (Previously presented) The method of claim 15 wherein the parameter specifies "sram Swap", and performing a swapping comprises swapping out a current context and waking it up when the thread's SRAM signal is received.

19. (Previously presented) The method of claim 15 wherein the parameter specifies "sram Swap", and performing a swapping comprises swapping a current context and waking it up when the thread's SDRAM signal is received.

20. (Previously presented) The method of claim 15 wherein the parameter specifies "inter_thread" which swaps out a current context and wakes it up when the thread's interthread signal is received.

21. (Original) The method of claim 15 further comprising:
an optional_token "defer one" which specifies that one instruction will be executed after this reference before the context is swapped.

22. (Currently amended) A parallel processor that can execute multiple contexts and that comprises:

- a register stack;
- a program counter for each executing context;
- an arithmetic logic unit coupled to the register stack and a program control store that stores a context swap instruction that causes the processor to:
 - evaluate a specified parameter specified in the context swap instruction to determine a state of an executing context process corresponding to a first thread; and
 - perform a swap operation to cause a different context and associated program counter, corresponding to a different thread, to be selected in accordance with the value of the evaluated specified parameter and which saves an old program counter value.

23. (Previously presented) The processor of claim 22 wherein the context swap instruction wakes up the swapped out context when a specified signal is activated.

24. (Currently amended) A computer program product residing on a computer readable medium for causing a multithreaded parallel processor to perform a function comprises instructions causing the processor to:

evaluate a specified parameter to determine a state of an executing context process
corresponding to a first thread; and

perform a swapping operation to cause a different context and associated program
counter, corresponding to a different thread, to be selected in accordance with the value of the
evaluated specified parameter.

25. (Currently amended) The product of claim 24 wherein the processor wakes up
the swapped out context when a specified signal corresponding to the specified parameter is
activated, wherein the specified parameter is identifies in a context-swap instruction.